Real-Time Surface Light-field Capture for Augmentation of Planar Specular Surfaces

Jan Jachnik* Imperial College London
Richard A. Newcombe† Imperial College London
Andrew J. Davison‡ Imperial College London

ABSTRACT

A single hand-held camera provides an easily accessible but potentially extremely powerful setup for augmented reality. Capabilities which previously required expensive and complicated infrastructure have gradually become possible from a live monocular video feed, such as accurate camera tracking and, most recently, dense 3D scene reconstruction. A new frontier is to work towards recovering the reflectance properties of general surfaces and the lighting configuration in a scene without the need for probes, omnidirectional cameras or specialised light-field cameras. Specular lighting phenomena cause effects in a video stream which can lead current tracking and reconstruction algorithms to fail. However, the potential exists to measure and use these effects to estimate deeper physical details about an environment, enabling advanced scene understanding and more convincing AR.

In this paper we present an algorithm for real-time surface light-field capture from a single hand-held camera, which is able to capture dense illumination information for general specular surfaces. Our system incorporates a guidance mechanism to help the user interactively during capture. We then split the light-field into its diffuse and specular components, and show that the specular component can be used for estimation of an environment map. This enables the convincing placement of an augmentation on a specular surface such as a shiny book, with realistic synthesized shadow, reflection and occlusion of specularities as the viewpoint changes. Our method currently works for planar scenes, but the surface light-field representation makes it ideal for future combination with dense 3D reconstruction methods.

Keywords: Real-Time, Light-Fields, Illumination Estimation, GPU, SLAM, AR

1 INTRODUCTION

Augmented Reality (AR) will surely have the potential for world-changing impact when the enabling technologies it relies on come fully together in low-cost, mass-market mobile devices. The more that can be done from the sensors built into these devices, and without the need for infrastructure, the more likely it is that designers will be able to create wide-reaching and generally useful AR applications. In reaction to this, there has been a strong move away from custom hardware in AR research, and interest has been particularly high in what can be done for AR from the video stream from a single moving camera.

Some of the most important steps along this path were taken by work on real-time SLAM using a single camera which was able to build self-consistent maps of features incrementally and use these for long-term, drift-free camera tracking. Davison *et al.*'s early MonoSLAM system [3] using sequential filtering was improved upon by other work such as [5] and most significantly by Klein and Murray's PTAM [7] with a parallel tracking and map optimisation approach which enabled greater tracking accuracy and dynamics of motion. In the past two years another big advance has been produced by the combination of modern optimisation algorithms and commodity parallel processing resources in the form of GPUs to permit real-time dense reconstruction from a single camera [11, 19, 13]. A dense surface model, generated live, allows AR objects to dynamically interact with the real scene; be occluded by, bounce off or even jump over real objects as shown in [11].

We consider the wealth of information in a standard real-time video stream from a moving camera which is currently still being ignored by most vision algorithms; and new parallel processing resources on GPU's give us the potential to deal with all of this data in real-time systems. Specifically, there is the potential to aim towards modelling of the reflectance properties of all scene surfaces and to map the full scene lighting configuration without any of the infrastructure such as lightprobes currently needed for such estimation. Newcombe *et al.*'s DTAM system [13] has paved the way for such an idea. DTAM creates dense 3D models using *every pixel* of the video feed. Expanding on this method, for each surface element in the scene, with estimated 3D position and normal, we can very rapidly gather a map of its appearance from different viewpoints as a hand-held camera browses. This is exactly the information which can then be used to infer both the reflectance properties of the surface, and the configuration of illumination in the scene.

The full problem of estimating lighting and reflectance properties for general scenes is surely a long-term one, with difficult joint estimation problems coming into view once issues such as complicated curved geometry and object inter-reflections and considered. In this paper we therefore make simplifying assumptions, but focus on an initial demonstration of real-time light-field capture, diffuse and specular lighting estimation and straightforward but highly effective placement of augmentations on specular surfaces — all with a single hand-held camera as the only data source. Specifically, we currently assume a static, planar scene; and static illumination which is well approximated as infinite relative to the camera motion. Like [13], we rely on the processing capability provided by a current commodity GPU, available as a normal gaming component in today's desktop and laptop computers, and design all of our algorithms to benefit from massive parallelism.

2 RELATED WORK

Much work on lighting and reflectance estimation has concentrated on single images, such as [6]. Nishino *et al.*[15] estimated the lighting and reflectance using a sparse set of images. At the other end of the scale [20] used a comparatively dense approach (several hundred images) to capture a view-dependent model of an object, but required a special capture rig. This information was then be used to estimate lighting and reflectance.

None of these approaches have taken advantage, as we do, of the huge amount of information coming from a 30fps live video feed; and in fact the processing and memory resources have only recently become widely available that makes dealing with such a large quantity of data feasible. The huge advantage of a real-time system is the feedback loop it gives to a user and we see in many

*e-mail: jrj07@doc.ic.ac.uk
†e-mail: r.a.newcombe@gmail.com
‡e-mail: ajd@doc.ic.ac.uk

AR systems which use vision for other capabilities. If processing is done post capture, we may find gaps where we decide that more data is needed, while a real-time user can immediately react to fill in these areas as appropriate.

Coombe *et al.* [1] use online singular value decomposition to capture a compressed representation of a surface light-field via principal component analysis. Although their method is memory efficient, it is only an approximation to the real light-field. Their system takes about 1 second to incorporate a new image in to the approximation. While interactive, it is not at frame-rate (30fps) and, hence, does not use every available piece of information.

The most closely related work to our own was by Davis *et al.*[2], who captured light-fields in real-time with an unstructured representation and use them for novel view rendering. Their system required only basic knowledge of scene geometry and stored the light-field as a series of keyframes. Novel views are then formed via interpolation of these keyframes. Although this unstructured representation has the advantage that it does not require an accurate 3D model, we believe that we can be more ambitious and that surface light-fields can offer much more information about surface properties, which can be used for BRDF estimation, material based segmentation and object detection. We envision combining the real-time 3D reconstruction algorithms already in existence [11, 12, 13] with real-time light-field capture.

Artificial reality relighting and shadows generally need light-probes or omni-directional cameras to capture the illumination distribution, such as [16]. Some methods aim to work without probes, but these are generally for only specialised cases. For example, Madsen and Lal [10] demonstrated probeless illumination estimation for outdoor scenes by considering the only light sources to be the sun and sky. Our work aims towards a real-time, probeless system which will work in any environment.

3 Algorithm Overview

The first stage in the algorithm is to capture a surface light-field. This step requires moving a handheld camera around the surface to observe it from different angles. For each frame we calculate the camera pose relative to the plane. A frame tells us what colour each visible surface element is from that particular viewpoint. We store this data in a surface light-field structure. All the processing is done on the GPU in real-time and is guided to ensure the user captures enough information. This step generally takes less than a minute.

The surface light-field is then split into two components: the view-independent diffuse part and the view-dependent specular part. The specular parts are merged together to create a global specular model of the surface. We can use this to estimate the environment map and generate shadows for virtual objects in AR. Virtual objects must occlude specularities to look realistic in a scene. The surface region which should be occluded is calculated and is filled with the diffuse colour, hence removing any specularities.

The 2 stages of light-field capture and viewing the augmented scene are disjoint. However, it could be possible to do these simultaneously, continually updating the light-field while viewing the augmented scene. However, due to the nature of the light-field capture this would still not work for dynamic lighting and so does not seem to bring any advantages.

4 Surface Light-field Representation

At the core of our approach is the capture of the light-field emanating from the surface of interest using a tracked single browsing camera. Rather than the more familiar 'two plane' light-field parameterisation [14, 8], we use a surface light-field as in [20]. A point on a specular surface reflects light with different colour and intensity depending on the viewpoint. Instead of a single RGB radiance value for a given surface texture element, we need to think of a radiance function $I(\omega)$ which is dependent on the viewing direction ω. A texture element can be seen from a hemisphere surrounding it. If $\mathbf{n}$ is the surface normal then $\omega \in \{\mathbf{s} \in S_2 \,|\, \mathbf{s} \bullet \mathbf{n} > 0\}$, where S_2 is the unit sphere. This radiance function $I(\omega)$, defined on the hemisphere, describes the colour of the texture element when viewed from any angle. We shall refer to this as a lumisphere (luminance + sphere, as in [20]). In this work we will concentrate only on planar surfaces ($\mathbf{n} = (0,0,1)$ for the whole surface). However, most of the concepts can be applied to a general surface as long as a geometric model of the surface, with normals, is known. This model could be captured with existing real-time methods such as [11, 12].

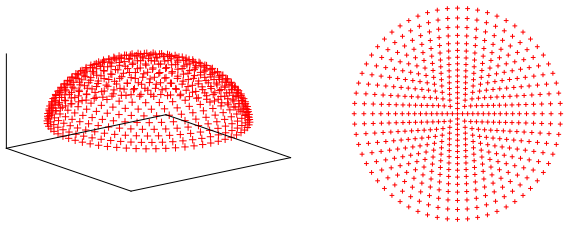


Figure 1: Discretisation of a hemisphere with $n = 16$. On the left is a 3D view. On the right is the stereographic projection onto the plane. Note that the points are evenly spread.

The surface light-field is represented as a 4D function $L(\mathbf{x}, \omega)$. In the planar case $\mathbf{x} = (x, y)$ is the 2D position of the texture element in a simple Cartesian coordinate system. For a general surface $\mathbf{x}$ will represent vertices on the surface. $\omega = (\theta, \phi)$ is the viewing direction. ω defines a point on a hemisphere with $\theta = 0$ as the north pole $\implies 0 \leq \theta < \pi/2$ and $0 \leq \phi < 2\pi$.

4.1 Hemisphere Discretisation

To represent the radiance function $I(\omega)$ we discretise the surface of a hemisphere and store a value for each discrete point. The samples on the hemisphere need to be as evenly spaced as possible to efficiently represent the distribution of outgoing light. The novel discretisation we use was inspired by the subdivided octahedral structure used in [20] but allows a much finer trade-off between resolution and memory requirement. The surface light-field is a four-dimensional structure and to get an adequate resolution on the hemisphere, memory becomes a serious issue.

We define a single integer parameter n which defines a number of discrete values $\theta \in \{\theta_0, ..., \theta_n\}$. This gives us a set of concentric circles on the surface of the sphere. Next, level set θ_i is evenly subdivided into $4i$ values for ϕ:

$$(\theta, \phi) \in \left\{ \left(\frac{i\pi}{2n}, \frac{2j\pi}{i} \right) \;\middle|\; j = 0, ..., i-1, \quad i = 0, ..., n \right\}. \qquad (1)$$

with $\phi = 0$ when $i = 0$. Figure 1 shows that the spread of the discrete points is relatively even across the surface of the hemisphere. The subdivision exhibits a triangular structure so interpolation can be performed by finding the 3 closest points. We typically choose $n \approx 60$ to give quite a high resolution on the hemisphere.

5 Reflection Model

We follow the majority of reflection models (e.g. Phong [18]) and assume that the light leaving our surface is the sum of two additive terms: a viewpoint-independent diffuse term and a view-dependent specular term. Having captured a whole lumisphere for a surface element, different methods have been tried to estimate the diffuse term. Nishino *et al.*[15] used the minimum observed value, and Wood *et al.*[20] the median of the observed values. We choose to use the median, because although it is computationally more expensive to compute it gives much more robustness in the case of

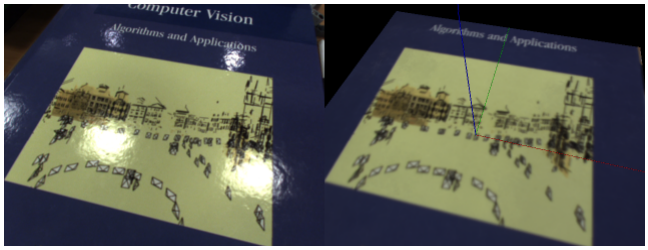


Figure 2: Live camera view (left) and the specularity-free diffuse texture (right) calculated from the medians of the lumispheres.

shadows or occasional bad camera tracking. During real-time light-field capture, it is possible to update the median of each lumisphere iteratively. To do this we store a histogram of the RGB values for each point on the surface and update this with each new frame. We also keep track of how many samples have contributed to the histogram. Given this information, calculating the median is as simple as finding in which bin the middle value lies.

The diffuse terms estimated for all surface elements can be combined into a diffuse texture for the surface, as illustrated in Figure 2. Since the specular radiance component is only visible from angles close to the direct mirror reflection of a light source, the specular lumisphere remaining once the diffuse component has been subtracted can be used for estimating the position of light sources. We will discuss this further in Section 7.

In practice, we do not decompose the light-field into its diffuse and specular parts iteratively as it does not seem necessary and requires more processing and memory. However, we do store and update the minimum for each surface element (very efficient and fast to calculate) as a guide to aid capture.

6 Data Capture

A calibrated camera with fixed exposure/shutter/gain browses a planar scene from multiple viewpoints. The pose of the camera is tracked by the freely available "Parallel Tracking and Mapping" (PTAM) program by Klein and Murray[7]. During initialisation, PTAM automatically defines a plane which is fitted to the observed features, and we take care that this aligns with the scene plane of interest for light-field capture. We also use the (x,y) coordinate system PTAM defines in this plane with the z-axis perpendicular to the plane. We should note that in order for PTAM to track the camera pose, the planar surface must have sufficient texture and we cannot yet cope with surfaces without significant diffuse texture.

Once tracking has been initialised, each incoming frame comes with a camera pose relative to the plane. Given the camera intrinsics we can then map world coordinates on the plane into image coordinates, as outlined in [9]. Note that for a diffuse surface, this is all that it needed to make a planar mosaic.

Given a new frame we back-project a ray from each point on the plane in to the camera image and use bilinear interpolation to give a pixel value. We use the pose of the camera to calculate the viewing direction for each pixel and calculate the 3 closest points on the lumisphere, using the geodesic distance. We then update these 3 points with a weight depending on the distance from the actual viewing direction and whether the point has been seen before.

Each discrete point on a lumisphere holds a 4 byte RGBA value. The alpha channel is simply used as a binary value. If the alpha value is zero then this means that the point has not been seen from that particular viewing direction.

6.1 Pixels to Irradiance

When measuring illumination we need to think in terms of the scene's radiance. Debevec and Malik[4] showed that radiance may map non-linearly to the pixel values given by a camera. Therefore, for our reflection model to hold true, we need to convert the pixel values from the camera into radiance values. For this, we need to know the camera response function (CRF), which can be obtained via the method outlined in [4]. The CRF translates pixel values to irradiance, the light incident on the sensor. We then assume that the radiance of the surface is proportional to the irradiance. Since we do not need to know absolute radiance values, only relative, it suffices to do our calculations in irradiance.

The method for finding the CRF is simple, and only needs to be done once for fixed camera settings. The camera position is fixed and images are captured at multiple, known shutter speeds. Pixels are then selected which span the full dynamic and colour range of the images. The values of these pixels, with their known shutter speeds, are combined into an optimisation problem which yields the CRF, for which a lookup table can be built for efficient lookup.

6.2 Feedback Mechanism

For specular surfaces we must use a high resolution lumisphere (at least 9000 discrete points) to give realistic results, since specularities are highly dependent on viewing direction. With each new video frame only one viewing direction is obtained per point on the surface, so it is unrealistic to expect to capture every viewing direction for every pixel.

We can make progress here by now stepping back from the completely general light-field we have captured, and recall the simplifying assumptions that our surface is planar, made of a constant material type, and that the illumination is dominantly from distant sources. This means that as long as we have seen at least one point on the surface from each viewing direction we can then use global surface/lighting properties to fill in the gaps (see Section 7). Given a camera with a wide angle lens, a single frame actually captures many viewing directions. This means capturing a view from each discrete direction is not time consuming. Our capture times are typically less than 1 minute for around 10% total coverage. From this sample global properties of the surface can be extracted well.

Our system incorporates a visual feedback mechanism (see Figure 3) to assist in getting coverage over the full range of viewing directions. This consists of a hemisphere which is coloured green when that particular viewing direction has been seen somewhere on the surface. The live viewing directions are shaded in red. This provides a feedback loop for the user to move the camera to previously unseen areas. The hemisphere is displayed to the user via stereographic projection on to the unit circle in the plane:

$$(r,\alpha) = \left(\tan\tfrac{\theta}{2},\phi\right) \tag{2}$$

where (r,α) are 2D polar coordinates. This stereographic projection is also used for visualisation of the lumispheres. Concentric circles are drawn to represent 10 degree intervals of the inclination angle θ. The aim is to try and fill the biggest circle fully, which means all viewing directions within 10 degrees of parallel to the surface have been seen. It is hard to capture data with angles shallower than 10 degrees, because tracking starts to fail.

6.3 Memory Usage

The surface light field is a four-dimensional structure and therefore consumes a vast amount of memory. We typically use a 256x256 grid for the planar surface. A high resolution light-field with 60 levels per lumisphere (7321 data points) takes up nearly 2GB of memory. Our current method simply allocates all the memory required. However, the captured light-field is highly sparse (typically $\sim 10\%$ fill) and would be well suited to a sparse quad-tree structure (for example). This would allow the memory usage to be compressed greatly. Current research hints towards efficient, scalable k-tree data-structures on GPU's with real-time update capabilities.

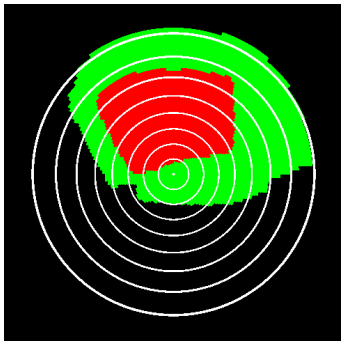


Figure 3: Visual feedback mechanism for light-field capture guidance. Red represents the current view, and green the coverage so far.

7 View-dependent Rendering

Given a full surface light-field we can construct an accurate view-dependent rendering of the surface from any viewing direction. The process of rendering the surface light-field is much like ray-tracing. Given the camera pose a ray can be drawn from the camera to each surface element, and used to calculate the viewing angle of each element. We then read the corresponding value from the lumisphere.

One big issue, as mentioned earlier, is the captured lumispheres are very sparse. It is clear that gaps in the data cannot be filled in realistically without some kind of global model. Without a global model, the best we can do is fill in the gaps with the diffuse colour. Figure 4(b) shows how artefacts appear when we fill in the gaps with diffuse data. From other viewing directions the specularities may not be visible at all, due to no data at those points.

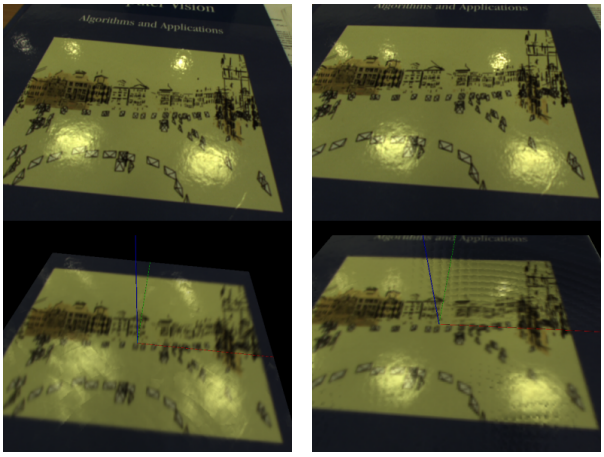
We tried two different ways to combat this problem: 1) Lower the resolution of the lumispheres and spend longer on the capture so that they are no longer sparse; 2) Use some kind of global model to fill in the gaps in the data. The first method is quite limited. Small changes in the viewing direction of a specular surface can result in large changes to the observation. A lower resolution means that these small changes are not detected and this results in spreading out of the specularities (Fig 4(a)). Quantisation effects are also evident when constructing the rendering from a low resolution. The high resolution light-field in Figure 4(b) shows much more realistic specularities but there are artefacts where there is missing data.

Method two, however, gives good results (see Figure 6). Our global model assumes that the planar surface is of the same material and, although it may have some varying diffuse texture, the specular component is consistent across the whole surface. This means that we can combine all of the sparse specular lumispheres into one non-sparse global specular lumisphere. Our visual feedback system ensures that this global lumisphere is filled. Figure 5 shows that the global specular lumisphere does not change significantly with extra data captured on top of what the feedback system deemed necessary. Clearly the feedback system works well.

Our assumption of globally constant specularities holds well. In the Phong reflectance model[18], for example, the specular term for a light source is given by:

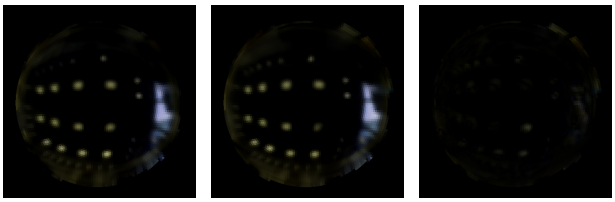
$$k_s(\cos\lambda)^{\alpha} i_s \tag{3}$$

where λ is the angle between the direction of the reflected light and the viewpoint; we assume the direction of the reflected light is constant across the surface, consistent with distant illumination. i_s is the intensity of the light source, so can readily be assumed to be constant. The exponent α is considered a property of the material and so can be assumed constant across surfaces of a single material. Specular reflection k_s is also assumed constant.



(a) Low resolution $n = 20$ (b) High resolution $n = 60$

Figure 4: Real camera views (top) and renderings from the light-field(bottom). The low resolution light-field tends to spread out specularities and has visible quantisation effects. The high resolution light-field is sparse and so there are gaps in the data where we can, at best, fill in with the diffuse colour



(a) 6% Fill (b) 20% Fill (c) Difference

Figure 5: The first image (a) has 6% fill of the light-field, just enough to cover all viewing directions as indicated by the feedback system. The second image (b) is after further capture to bring it up to 20% fill. (c) is a difference image. The mostly noticeable difference is that one of the lights is dimmer in the 20% fill image. This is in fact due to the user occluding that light during further capture. There are no major differences which shows that our feedback mechanism works well to obtain just the required amount of information.

The assumption that k_s is constant across the surface is possibly unrealistic because we would expect that a blue part of the surface will reflect more blue light than a red part of the surface. However, in practice, our assumption gives good results with no apparent skew of colours. There is the possibility that k_s could be related to the diffuse colour of the surface, which we have already calculated. This is an area which requires further research. The surfaces we consider are of the same material but the colour may vary (e.g. the surface of a book). The distance to light sources is several times bigger than the size of the surface so the distant illumination assumption is approximately true.

7.1 Results

Figure 6 shows a comparison of real camera views with views rendered from the captured light-field with the global model. Table 1 shows some quantitative results obtained from these comparisons. We see that we have captured the general size, colour (hue) and position of the major light sources. Due to the averaging over the lumispheres, the fine texture of the surface (in a and b) is not captured. The surface is not truly planar. This fine texture is only visible at specularities so capturing this texture is difficult, and would

(a) (b) (c)

Figure 6: Comparison of real camera views (top) and views rendered from the light-field (2nd row) using the global specular lumisphere. The 3rd row shows absolute difference images and the bottom row shows difference of gradient images. The position and colour of the specularities is captured well but the intensity is underestimated. The global model smooths the fine texture in (a) and (b) but is not an issue in (c), clearly shown by the difference of gradients. The surface in (a) and (b) has a dappled texture which is not correctly modelled by the planar assumption.

presumably require high resolution per-surface-element normal estimation (bump-map capture) — something we could certainly extend our approach to attempt in the future.

The images show that the intensity of the specularities appear to be underestimated. There are 2 things which contribute to this. The first is saturated pixels during capture. The table shows what percentage of pixels are saturated in the live image, as a guide. It is clear that the largest errors are in image (b). In this case, not only is the specular component not reproduced accurately but the diffuse component has a larger error than the other examples. The saturation during capture has skewed the decomposition into diffuse and specular components. To avoid these errors, the exposure of the camera must be chosen carefully to avoid saturated pixels. However, in some cases we are limited by the dynamic range of the camera but this can easily be overcome.

The second factor which affects the intensity is using the median to estimate the diffuse component. This will tend to give a diffuse estimate slightly brighter than the real value, dampening the effect of the specular component. While the minimum may give a more realistic estimate of the diffuse component, it is not robust. Most noticeably, when using the minimum, slight errors in camera pose cause visible shrinking of the borders of bright surface regions.

7.2 Environment Map Approximation

We note that the calculated global lumisphere can be used as an estimate of the environment map and used to predict the positions of light sources. Figure 7 compares the global lumispheres we capture with our method with environment maps using the standard reflective sphere 'probe' method (hence the reflection of the cameraman). The light-fields were captured from different surfaces (both shiny books) and in different rooms. Both show good estimates of the environment map, picking up the colours of the lights and windows. It is possible to see some of the green of a tree outside the window.

Nowrouzezahrai *et al.*[16] used an environment map to apply realistic shadows to augmented reality objects. They applied a spherical harmonic approximation to the environment map to greatly reduce the resolution and enable effective separation of light sources to cast hard and soft shadows. We believe that our specular lumisphere method captures an approximation of the environment map of sufficient quality which is suitable for such shadow effects, without the need for light probes or other capture devices.

Image	RMSE Intensity	Max Grad. Diff.	Saturation
(a)	0.0489	0.765	0.02%
(b)	0.1160	1.000	2.90%
(c)	0.0292	0.247	0.00%

Table 1: Quantitative evaluation of the difference between real camera views and views rendered using the global specular lumisphere modelling using light-field capture, for the five images (a)–(c) shown in Figure 6. We show RMSE normalised image intensity difference, normalised maximum gradient difference, and the percentage of saturated pixels in each real image. We observe that good general agreement between real and rendered views is obtained but that, with the current method, differences get larger in images with significant saturation. It is also interesting to note that the gradient difference measure reveals something extra, that we get better agreement for image (c) where the surface is smooth and lacks the fine dappled texture of the book used in (a) and (b). Our method currently assumes a perfectly planar surface.

8 Implementation

Camera pose tracking by PTAM is implemented on the CPU. When a new frame is grabbed, it is sent to PTAM to calculate a pose and then copied to the GPU (NVIDIA GeForce GTX580 GPU with 3GB RAM) for all further processing. The light-field data structure is stored and updated entirely on the GPU. The fact that the problem is highly parallel and the power of GPU's for parallel processing enables our capture and rendering systems to run in real-time. Excluding PTAM, the system takes around 6-7ms to capture or render the light-field.

8.1 Irradiance from Pixel Values

The camera used is a Point Grey Flea2 with a wide angle lens. We have measured it to have a CRF which is highly linear up to close to saturation point and we leave the shutter time constant. Therefore, in fact, no conversion between pixel values and irradiance is necessary as long as we choose our exposure settings to avoid saturated pixels during light-field capture — image brightness is proportional to irradiance. If for a different camera the mapping was significantly non-linear we would simply use the CRF to map the pixel values to radiance values. If pixels become saturated then the decomposition into diffuse and specular terms becomes invalid. This means that our capture is limited by the dynamic range of the camera. However, this could easily by overcome by dynamically changing the shutter speed to capture a wider dynamic range.

9 Augmented Reality on Specular Surfaces

A huge issue with augmented reality on specular surfaces is that virtual objects occlude specularities. Therefore, these specularities need to be removed to get a realistic AR view. Figure 8 shows how important this is for making AR look realistic.

The surface light field representation gives us the ability to remove the specularities. There are two ways of doing this. The first

Figure 8: Augmented scenes with and without illumination, shadows, occlusion and reflection. The views with the effects are much more convincing. While augmented reality is generally performed on diffuse surfaces, our method brings realistic AR to specular surfaces. The most obviously important aspect for realism is the occlusion of the specularities by virtual objects.

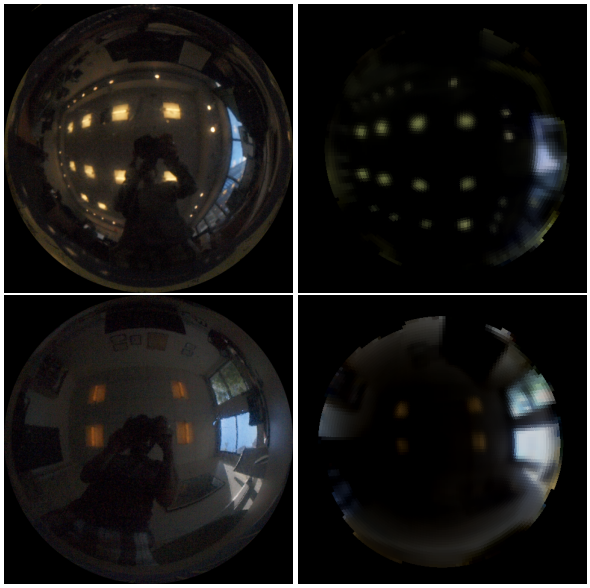


Figure 7: Environment maps for two different rooms captured using a probe (left) and using our method based on specular reflections from a shiny book (right). The direction, intensity and colour of the light sources is captured well.

is to extract the specular component from the light-field and subtract this from the incoming video feed. However, we found that our specular model is not yet accurate enough to produce good results with this approach. The second method involves replacing the occluded region with the diffuse term, as calculated from the surface light-field. This gives very promising results. Figure 9 shows how a virtual object occludes the specularities. In the occluded region we have combined the surface's diffuse component and the reflection of the virtual object.

The occlusion area is calculated by a simple ray-object intersect method which is implemented on the GPU in 5-6ms.

9.1 Shadows

Virtual shadows are added via a shadow map. For each point on the surface, we estimate what proportion of radiance from the captured environment map is occluded by the virtual object. The intensity of the shadow is then calculated to be proportional to this value. This is a brute-force approach, and i mplemented on the GPU takes 0.3 seconds, but with some optimisation should run at frame-rate to enable dynamic shadow generation for moving AR objects.

An alternative approach to add shadows is the method outlined in [16]. They pick a dominant light source to give hard shadows and use a spherical harmonic representation of the environment map to calculate soft shadows. This method could easily be applied to our system, using the estimated environment map.

Note that saturated pixels can cause visual artefacts when blending shadows with a video feed - as seen in the middle images of figure 9. Given that there is no measure of how saturated a pixel is, there is no correct way to apply shadows to these saturated regions.

9.2 Illumination

Illuminating the virtual object is a two-step process. So far we consider only diffuse virtual objects. First we need to calculate the illumination of the object with respect to the environment map. Then we need to calculate the illumination due to the surface. Since the surface is specular, the illumination effects on the object will be direction dependent. We are yet to implement this full reflectance model but we have done some basic relighting based on dominant light sources given by the environment map. This is used in the examples and gives satisfactory results.

9.3 Results

The results we present below are also shown in an extended form in the video we have submitted alongside the paper. The video also shows the different stages of how the system works. Video URL: `http://www.doc.ic.ac.uk/~jrj07/ismar2012.html`

Figures 8 and 9 show the results of shadows, specular occlusion and basic illumination for a variety of surfaces and environments. These effects clearly make the virtual object look more realistically placed in the scene. For the glass-fronted picture on the far right of Figure 8, the specular reflection is almost mirror-like and this makes the reflection of the object look very realistic. On surfaces with less mirror-like reflections (like the books) the direct reflection looks slightly out of place because it should be diffused by the surface. The information on how this reflection should diffuse is contained in the surface light-field so we hope to improve this in the future.

10 Further Work

Currently the camera browsing the scene is restricted to have a constant shutter time. For this proof-of-concept application this is adequate, but for real scenes a wider dynamic range is needed. This constraint can easily be relaxed if we know the shutter time and camera response function so that we can convert pixel values to radiance values.

A clear next step with this work is to seek to combine light-field capture with live dense reconstruction as in [13]. There are many surfaces in the real world (such as the books we have used in our experiments) which have significant texture but also partial specular reflection characteristics. The shape of such surfaces can be reconstructed in 3D based on their diffuse texture, and then a surface light-field can be defined relative to this shape for the capture of specular lighting effects. A longer term challenge is to deal with objects whose surfaces are more purely specular, and do not offer enough texture for standard stereo matching and reconstruction. Coping with these will involve substantial future work on the joint estimation of surface shape and reflectance properties.

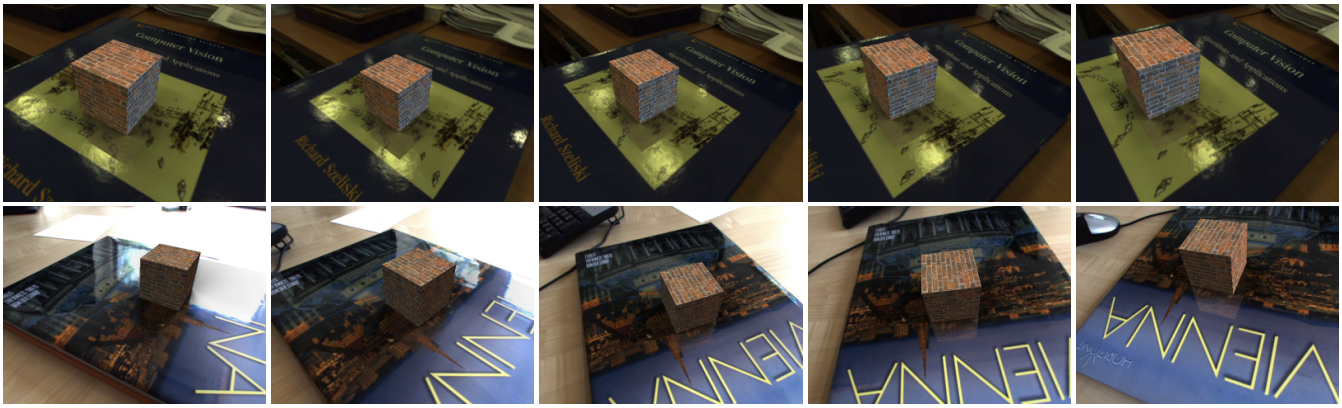


Figure 9: Sequence shots from two augmented video sequences. Notice how the object occludes the reflection of the light. The reflection is combined with the diffuse texture so that it joins seamlessly. Please see our video to view these clips, all rendered in real-time.

We hope to try surface light-field capture with a plenoptic camera[14]. The amount of data captured on each frame with a light-field camera is far greater than that of a normal camera. This means we can fill the surface light-field structure more densely. Additionally, it is possible to generate depth maps from light-field cameras[17]. This could be a step towards simultaneous dense reconstruction and surface light-field capture.

Finally, if we capture an environment map and a surface light-field, we know both the incoming and outgoing light at the surface. This should lead to good BRDF estimation. This could then be used for material based segmentation to aid object recognition.

11 Conclusion

We have demonstrated a new method for recovering detailed lighting and surface information with a standard single hand-held camera AR configuration. In standard monocular tracking and mapping systems the effects caused by complicated lighting and specular surfaces are ignored or even cause problems. These can, in fact, be used for estimation of properties crucial for realistic AR. Specifically, in this paper we have demonstrated easy and rapid light-field capture from planar specular surfaces, and the straightforward use of this for reflectance and environment map estimation without the need for any additional probes or hardware. We have given convincing real-time demonstrations of the placement of augmentations of specular surfaces with realistically synthesized reflections, shadows, illumination and specularity occlusion.

Acknowledgements

This work was supported by an EPSRC DTA Scholarship to J. Jachnik, and European Research Council Starting Grant 210346. We are extremely grateful to members of Imperial College's Robot Vision Group for day to day help and collaboration.

References

[1] G. Coombe, C. Hantak, A. Lastra, and R. Grzeszczuk. Online Construction of Surface Light Fields. In *Eurographics Symposium on Rendering*, 2005.

[2] A. Davis, M. Levoy, and F. Durand. Unstructured Light Fields. In *Eurographics*, 2012.

[3] A. J. Davison, W. W. Mayol, and D. W. Murray. Real-Time Localisation and Mapping with Wearable Active Vision. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2003.

[4] P. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *ACM Transactions on Graphics (SIGGRAPH)*, 1997.

[5] E. Eade and T. Drummond. Unified Loop Closing and Recovery for Real Time Monocular SLAM. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2008.

[6] K. Hara, K. Nishino, and K. Ikeuchi. Light Source Position and Reflectance Estimation from a Single View without the Distant Illumination Assumption. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27:493–505, 2005.

[7] G. Klein and D. W. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

[8] M. Levoy and P. Hanrahan. Light Field Rendering. In *ACM Transactions on Graphics (SIGGRAPH)*, 1996.

[9] S. J. Lovegrove. *Parametric Dense Visual SLAM*. PhD thesis, Imperial College London, 2011.

[10] C. Madsen and B. Lal. *Probeless Illumination Estimation for Outdoor Augmented Reality*. INTECH, 2010.

[11] R. A. Newcombe and A. J. Davison. Live Dense Reconstruction with a Single Moving Camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[12] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[13] R. A. Newcombe, S. Lovegrove, and A. J. Davison. DTAM: Dense Tracking and Mapping in Real-Time. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[14] R. Ng, M. Levoy, M. Bredif, G. Duval, M. Horowitz, and P. Hanrahan. Light Field Photography with a Hand-held Plenoptic Camera. Technical report, Stanford Tech Report CTSR, 2005.

[15] K. Nishino, Z. Zhang, and K. Ikeuchi. Determining Reflectance Parameters and Illumination Distribution from a Sparse Set of Images for View-dependent Image Synthesis. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2001.

[16] D. Nowrouzezahrai, S. Geiger, K. Mitchell, R. Sumner, W. Jarosz, and M. Gross. Light factorization for mixed-frequency shadows in augmented reality. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 173–179, 2011.

[17] C. Perwass and L. Wietzke. Single Lens 3D-Camera with Extended Depth-of-Field. In *SPIE*, volume 8291, 2012.

[18] B. T. Phong. Illumination for Computer Generated Pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[19] J. Stuehmer, S. Gumhold, and D. Cremers. Real-Time Dense Geometry from a Handheld Camera. In *Proceedings of the DAGM Symposium on Pattern Recognition*, 2010.

[20] D. Wood, D. Azuma, W. Aldinger, B. Curless, T. Duchamp, D. Salesin, and W. Stuetzle. Surface Light Fields for 3D Photography. In *ACM Transactions on Graphics (SIGGRAPH)*, 2000.